Topic 3: Sentiment Analysis

Clarissa

2022-04-18

"IPCC" Nexis Uni data set sentiment plot

```
IPCC_files <- list.files(pattern = "Nexis_IPCC_Results.docx", path = here::here("data"),
                         full.names = TRUE, recursive = TRUE, ignore.case = TRUE)

dat_IPCC <- lnt_read(IPCC_files) # Object of class 'LNT output'

# split LNT output class into three different dfs
meta_df_IPCC <- dat_IPCC@meta
articles_df_IPCC <- dat_IPCC@articles
paragraphs_df_IPCC <- dat_IPCC@paragraphs

dat2_IPCC <- data_frame(element_id = seq(1:length(meta_df_IPCC$Headline)),
                        Date = meta_df_IPCC$Date,
                        Headline = meta_df_IPCC$Headline)
```

```
## Warning: `data_frame()` was deprecated in tibble 1.1.0.
## Please use `tibble()` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was generated.
```

```
# can we create a similar graph to Figure 3A from Froelich et al.?
mytext_IPCC <- get_sentences(dat2_IPCC$Headline)

# approximate the overall sentiment for a given text (scale -1 to 1)
# (attempts to correct for negation, context, etc.)
sent_IPCC <- sentiment(mytext_IPCC)

sent_df_IPCC <- inner_join(x = dat2_IPCC, y = sent_IPCC,
                           by = "element_id")

sentiment_IPCC <- sentiment_by(sent_df_IPCC$Headline)

sent_df_IPCC %>%
  arrange(sentiment)
```

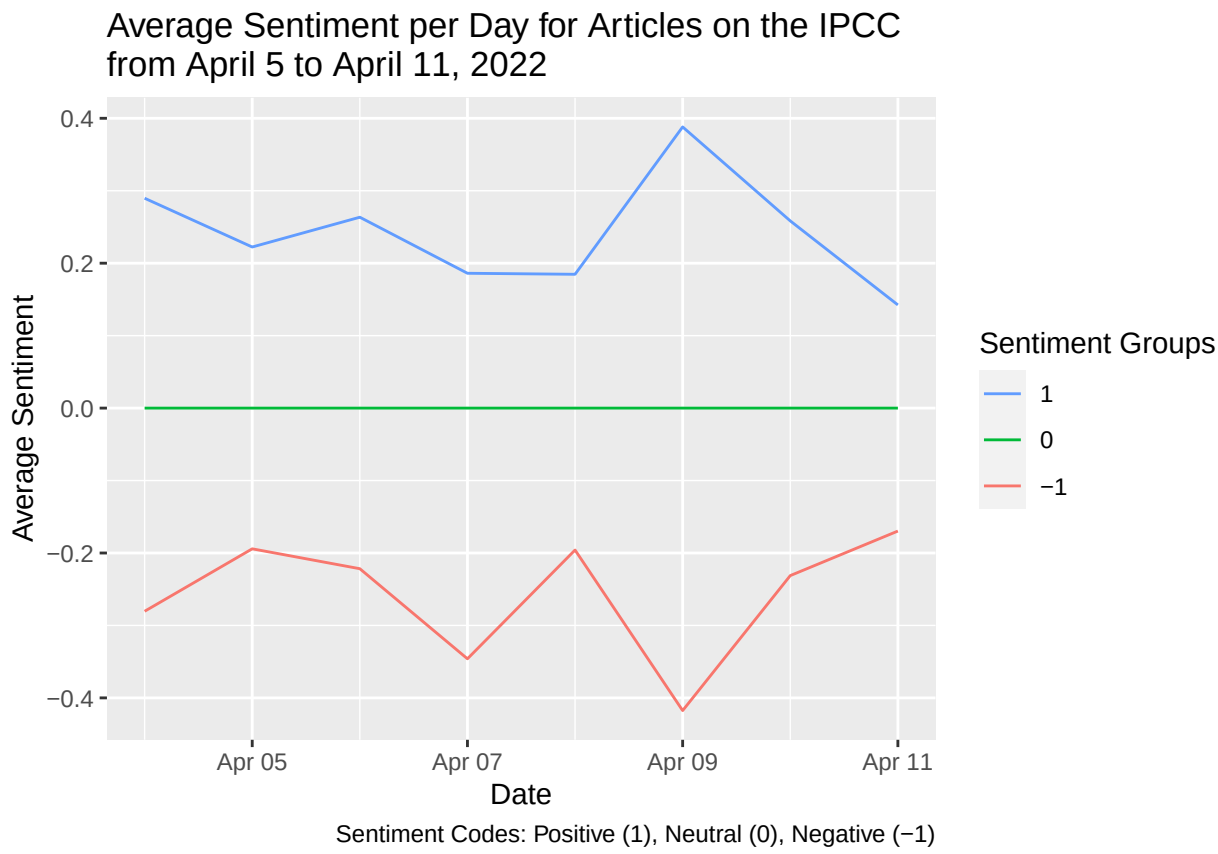
```
## # A tibble: 109 x 6
##    element_id Date       Headline                sentence_id word_count sentiment
##         <int> <date>     <chr>                         <int>      <int>     <dbl>
```

```
##  1         66 2022-04-04 Scientists risk arres~           1          7    -0.756
##  2         91 2022-04-07 The 'climate change' ~           1          9    -0.75
##  3         28 2022-04-09 The Dread 1.5 Degree ~           1          6    -0.714
##  4         43 2022-04-06 India's banks unprepa~           1          7    -0.510
##  5         34 2022-04-08 Dangerous radicals ar~           1          6    -0.449
##  6         14 2022-04-04 'Now or never' to avo~           1          8    -0.442
##  7         78 2022-04-07 Statewide Gas Ban Bil~           1         10    -0.427
##  8         50 2022-04-04 Guardian: Media 'Bare~           1          8    -0.407
##  9         62 2022-04-06 Governor Youngkin's I~           1         11    -0.377
## 10          7 2022-04-05 Narrow path to avoid ~           1          8    -0.354
## # ... with 99 more rows
```

```
sent_df_IPCC %>%
  mutate(sentiment_groups = case_when(sentiment > 0 ~ "1",
                                      sentiment == 0 ~ "0",
                                      sentiment < 0 ~ "-1"),
         factor(sentiment_groups, levels = c(1, 0, -1))) %>%
  group_by(Date, sentiment_groups) %>%
  summarise(mean_sentiment = mean(sentiment)) %>%
  ggplot(aes(x = Date,
             y = mean_sentiment,
             color = sentiment_groups)) +
  geom_line(position = "dodge") +
  labs(col = "Sentiment Groups",
       y = "Average Sentiment",
       title = "Average Sentiment per Day for Articles on the IPCC \nfrom April 5 to April 11, 2022",
       caption = "Sentiment Codes: Positive (1), Neutral (0), Negative (-1)") +
  guides(color = guide_legend(reverse = TRUE))
```

```
## `summarise()` has grouped output by 'Date'. You can override using the `.groups` argument.
```

```
## Warning: Width not defined. Set with `position_dodge(width = ?)`
```



"Heat Related Death" Nexis Uni data set

```
my_files_heat <- list.files(pattern = "Files_100_heat_related_death.docx", path = here::here("data"),
                            full.names = TRUE, recursive = TRUE, ignore.case = TRUE)

dat_heat <- lnt_read(my_files_heat) # Object of class 'LNT output'
```

```
## Creating LNToutput from 1 file...

##  ...files loaded [0.084 secs]

##  ...articles split [0.095 secs]

##  ...lengths extracted [0.098 secs]

##  ...headlines extracted [0.099 secs]

##  ...newspapers extracted [0.10 secs]

##  ...dates extracted [0.11 secs]

##  ...authors extracted [0.11 secs]
```

```
##  ...sections extracted [0.11 secs]

##  ...editions extracted [0.11 secs]

## Warning in lnt_asDate(date.v, ...): More than one language was detected. The
## most likely one was chosen (English 87%)

##  ...dates converted [0.12 secs]

##  ...metadata extracted [0.12 secs]

##  ...article texts extracted [0.12 secs]

##  ...superfluous whitespace removed [0.13 secs]

## Elapsed time: 0.13 secs
```

```
# split LNT output class into three different dfs
meta_df_heat <- dat_heat@meta
articles_df_heat <- dat_heat@articles
paragraphs_df_heat <- dat_heat@paragraphs

dat2_heat <- data_frame(element_id = seq(1:length(meta_df_heat$Headline)),
                        Date = meta_df_heat$Date,
                        Headline = meta_df_heat$Headline)

paragraphs_dat_heat <- data_frame(element_id = paragraphs_df_heat$Art_ID, Text  = paragraphs_df_heat$Pa

dat3_heat <- inner_join(dat2_heat, paragraphs_dat_heat, by = "element_id")
```

```
cleaned_data_heat <- dat3_heat %>%
  mutate(text_https = str_detect(string = dat3_heat$Text, pattern = "https", negate = TRUE)) %>%
  filter(text_https == TRUE)

nrc_sentiment <- get_sentiments('nrc') #grab the bing sentiment lexicon from tidytext
head(nrc_sentiment, n = 20)
```

```
## # A tibble: 20 x 2
##    word        sentiment
##    <chr>       <chr>
##  1 abacus      trust
##  2 abandon     fear
##  3 abandon     negative
##  4 abandon     sadness
##  5 abandoned   anger
##  6 abandoned   fear
##  7 abandoned   negative
##  8 abandoned   sadness
##  9 abandonment anger
## 10 abandonment fear
## 11 abandonment negative
```

```
## 12 abandonment sadness
## 13 abandonment surprise
## 14 abba        positive
## 15 abbot       trust
## 16 abduction   fear
## 17 abduction   negative
## 18 abduction   sadness
## 19 abduction   surprise
## 20 aberrant    negative
```

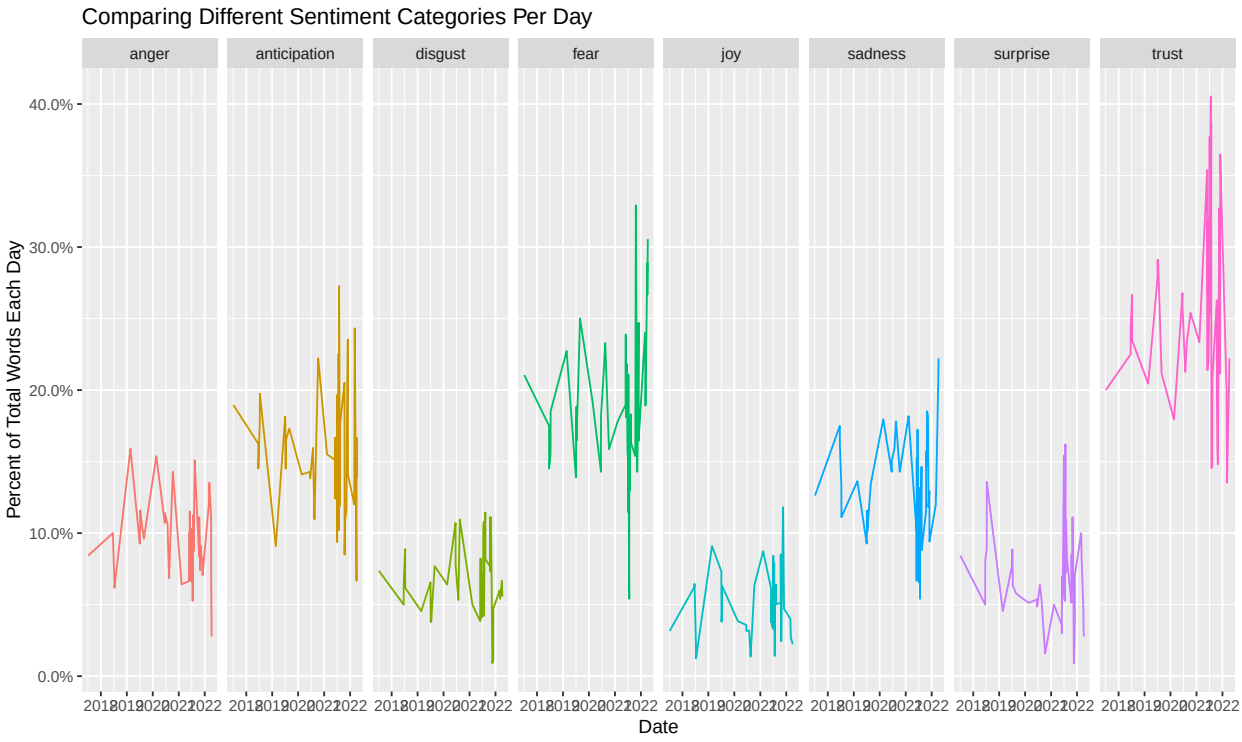
```
cleaned_data_heat_words <- cleaned_data_heat  %>%
  select(!text_https) %>%
  unnest_tokens(output = word, input = Text, token = 'words')

cleaned_data_heat_sentiment_words <- cleaned_data_heat_words %>% #break text into individual words
  anti_join(stop_words, by = 'word') %>% #returns only the rows without stop words
  inner_join(nrc_sentiment, by = 'word') %>% #joins and retains only sentiment words
  filter(!sentiment %in% c("negative", "positive"))
```

```
data_heat_graph <- cleaned_data_heat_sentiment_words %>%
  group_by(Date, sentiment) %>%
  summarise(count = n()) %>%
  mutate(sum_count = sum(count))
```

```
## `summarise()` has grouped output by 'Date'. You can override using the `.groups` argument.
```

```
ggplot(data = data_heat_graph,
       aes(x = Date,
           y = count / sum_count,
           color = sentiment)) +
  geom_line() +
  scale_y_continuous(labels = percent) +
  facet_grid(~sentiment) +
  theme(legend.position = "none") +
  labs(y = "Percent of Total Words Each Day",
       title = "Comparing Different Sentiment Categories Per Day")
```

```
## Warning: Removed 8 row(s) containing missing values (geom_path).
```




```
highest_trust <- cleaned_data_heat_sentiment_words %>%
  filter(sentiment == "trust") %>%
  group_by(word) %>%
  summarise(count = n())
```